Harnessing the Power of PDF

Session 213

These are confidential sessions—please refrain from filming or taking pictures.

Graphics and Media

Harnessing the Power of PDF

Paul Danbold
Imaging Technologies Evangelist

http://developer.apple.com/wwdc2005/

What You'll Learn

- Why Apple likes PDF
- All the ways in which we support PDF
- How you can support PDF

What Is PDF?

http://developer.apple.com/wwdc2005/

“PDF is an extensible form of paper, a hypermedia that is device independent, platform independent, color consistent, and it is the best universal transmission media for creative and intellectual assets.”

John Warnock, co-founder Adobe Systems, Inc.

Why Apple Likes PDF

http://developer.apple.com/wwdc2005/

PDF Is…

- A proven technology
- A public specification
- Platform independent
- A compact file format
- A secure file format

PDF as Digital Paper

- Resolution and device independent
- Page-oriented
- Searchable
- Encryptable
- Annotatable
- Ideal for printing, sharing, archiving

What You Can Do With PDF

http://developer.apple.com/wwdc2005/

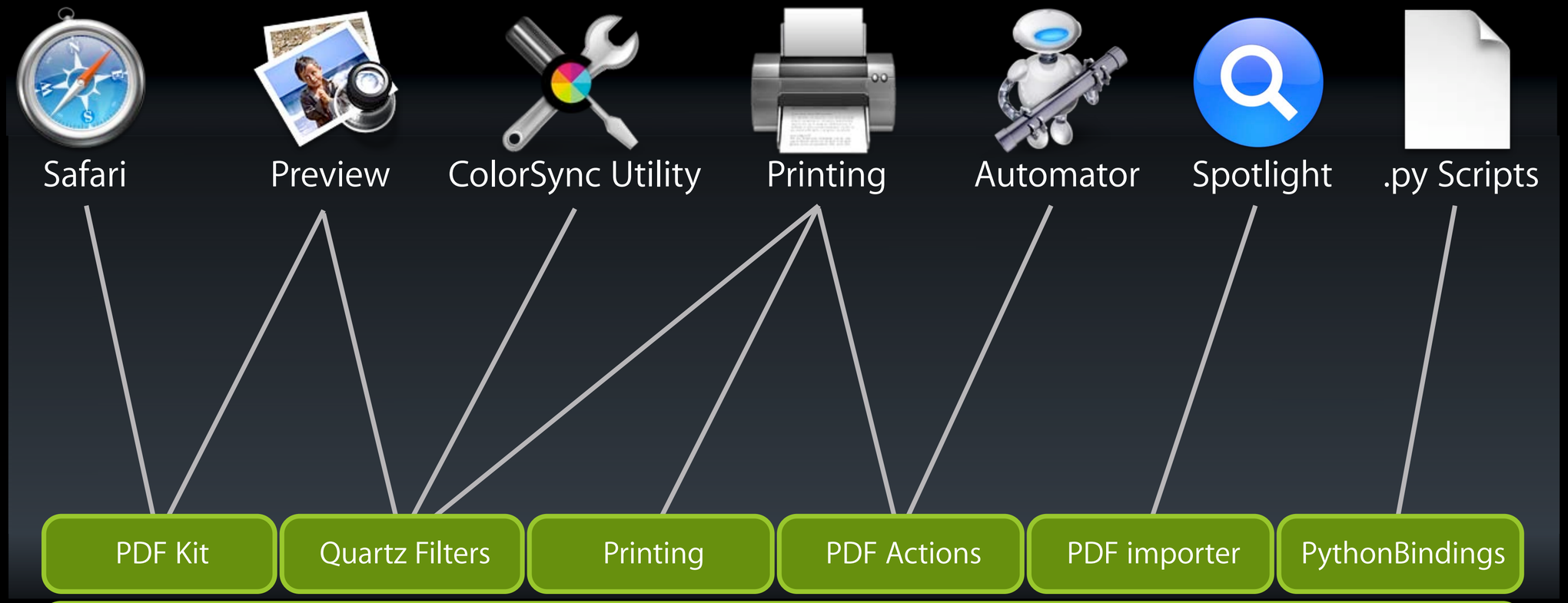
With Little Effort, Your Application Can…

- Open and display PDF documents
  - From files and from the pasteboard
- Generate PDF files
  - From the Print dialog
  - Via your Save As or Export menus
  - Put PDF on the pasteboard
- Modify and annotate PDF files
- Use PDF as a solution for handling legacy graphics formats
  - EPS, PS, PICT
- Participate in PDF workflows
  - PDF Workflow tools
  - Automator actions
  - Python scripts

PDF Services in Mac OS X

http://developer.apple.com/wwdc2005/

PDF Everywhere



Quartz 2D

The lowest level APIs for reading, displaying, generating PDF

Quartz 2D Support for PDF

A complete set of APIs to

- Open PDF documents
  - `CGPDFDocumentCreateWithURL()`
- Obtain PDF from the pasteboard
  - `CGPDFDocumentCreateWithDataProvider()`
- Handle password-protected PDF files
  - `CGPDFDocumentUnlockWithPassword()`
- Draw a PDF page
  - `CGContextDrawPDFPage()`
- Get information about the PDF document
  - `CGPDFDocumentGetNumberOfPages()`, `CGPDFPageGetBoxRect()`, `CGPDFDocumentGetCatalog()`, etc.

Quartz 2D Support for PDF

A complete set of APIs to

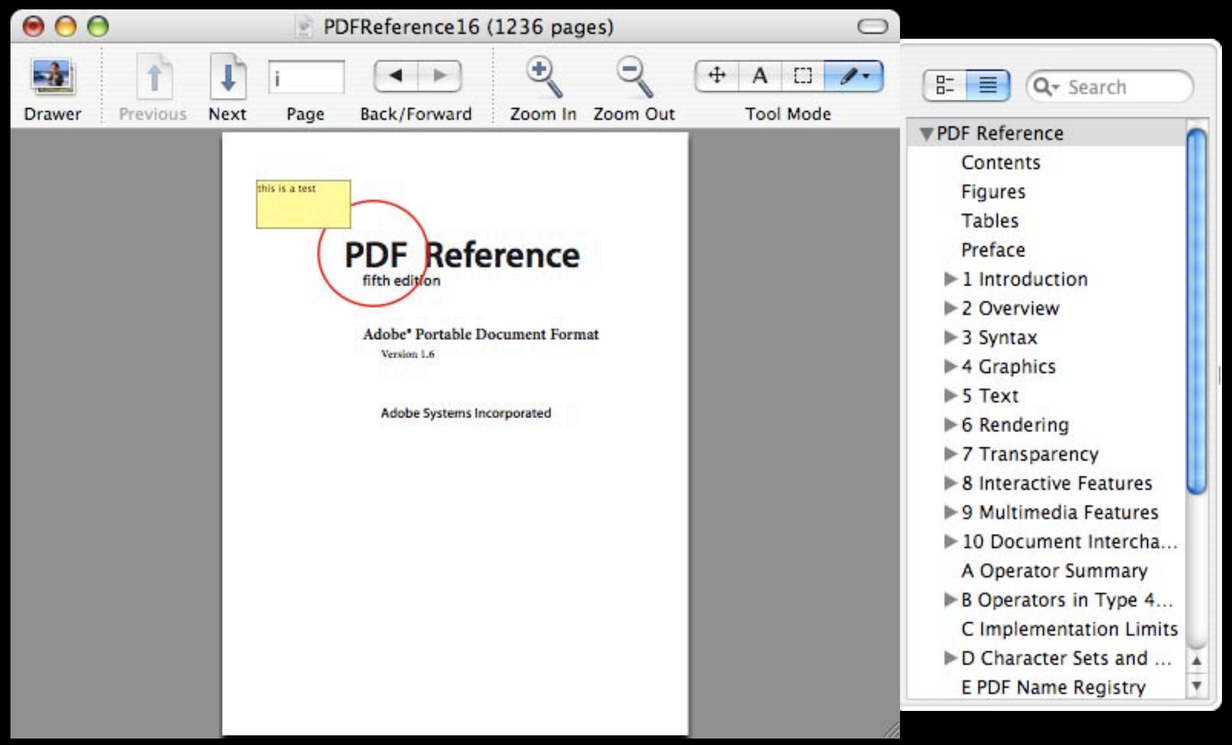
- Create PDF documents
  - `CGPDFContextCreateWithURL()`
- Copy PDF documents to the pasteboard
  - `CGPDFContextCreate()` with a custom data consumer
- Add links to a PDF document (new in Tiger)
  - `CGPDFContextAddDestinationAtPoint()` with `CGPDFContextSetDestinationForRect()`
- Access the PDF content stream (new in Tiger)
  - `CGPDFScannerScan()`

Quartz 2D and the PDF Specification

- Quartz and the PDF specification
  - In Tiger, we can read and draw PDF 1.6
  - PDF generated via Quartz starts at PDF 1.3

PDF Kit

Easy PDF support for Cocoa apps on Tiger

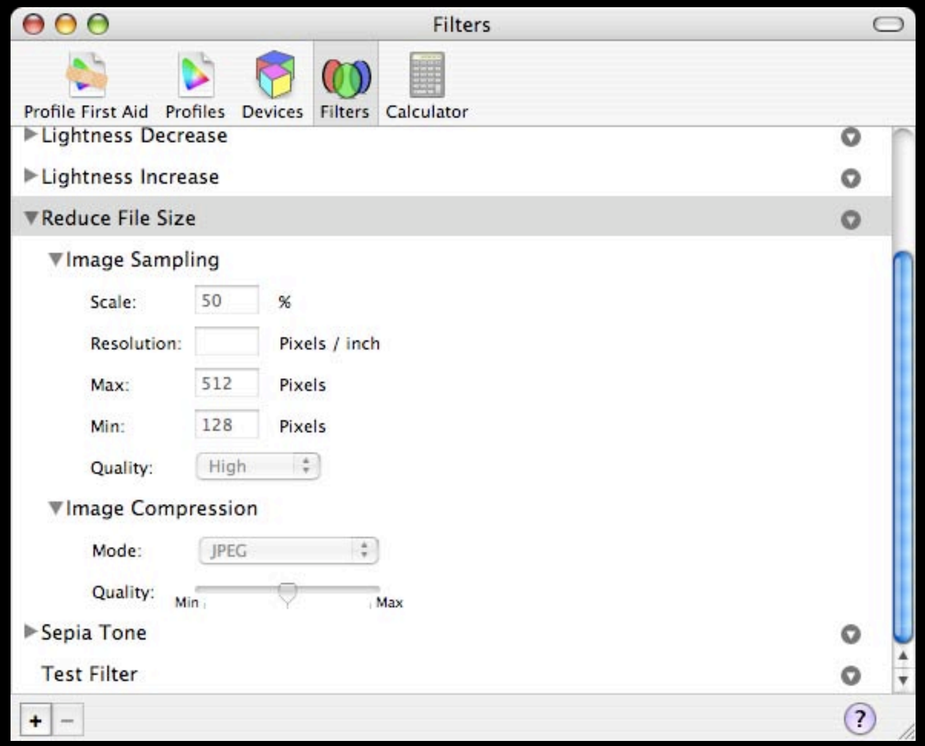


PDF Kit | Quartz Filters | Printing | PDF Actions | PDF importer | PythonBindings

Quartz 2D

Quartz Filters

Post-processing of PDF files

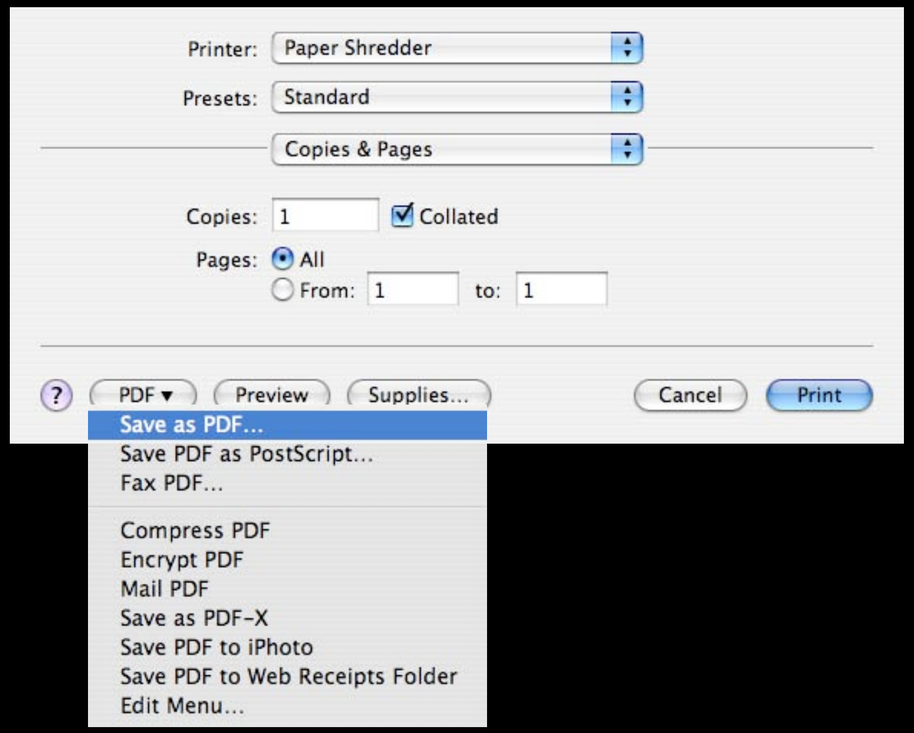


PDF Kit | **Quartz Filters** | Printing | PDF Actions | PDF importer | PythonBindings

Quartz 2D

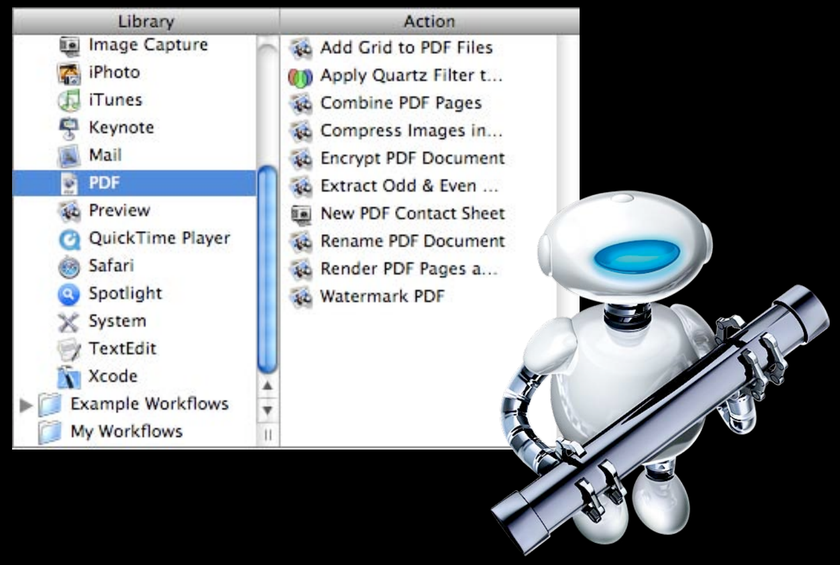
PDF and the Printing System

PDF creation for all applications



PDF Actions

PDF support for Automator workflows



PDF Kit | Quartz Filters | Printing | **PDF Actions** | PDF importer | PythonBindings

Quartz 2D

Python Scripts

PDF processing with Python

Terminal — bash — 92x31

```
from CoreGraphics import *
import sys, os, math, getopt, string

def usage ():
  print '''
usage: python watermark.py [OPTION]... INPUT-PDF OUTPUT-PDF

Add a "watermark" to a PDF document.

  -t, --text=STRING
  -f, --font-name=FONTNAME
  -F, --font-size=SIZE
  -c, --color=R,G,B
'''

def main ():

  text = 'CONFIDENTIAL'
  color = (1, 0, 0)
  font_name = 'Gill Sans Bold'
  font_size = 36
  page_rect = CGRectMake (0, 0, 612, 792)

try:
  opts,args = getopt.getopt (sys.argv[1:], 't:f:F:c:',
                             ['text=', 'font-name=', 'font-size=', 'color='])
except getopt.GetoptError:
  usage ()
  sys.exit (1)

if len (args) != 2:
```

PDF Kit | Quartz Filters | Printing | PDF Actions | PDF importer | PythonBindings

Quartz 2D

Dealing With Legacy Graphic Formats

PostScript, Encapsulated PostScript, QuickDraw PICT

- PS and EPS
  - Carbon—`CGPSConverterCreate(), CGPSConverterConvert()`
  - Cocoa—`NSEPSImageRep`
  - BSD—`/usr/bin/pstopdf`
- PICT
  - Carbon—`QDPictXXX()`
  - Cocoa—`NSImage`

Summary, So Far

- Use the PDF functions in Quartz 2D for the most comprehensive and lowest level PDF services
- At a higher level, PDF Kit is ideal for Cocoa apps on Tiger
- For PDF Workflows, look at Quartz Filters, Automator PDF Actions, and Python Scripts

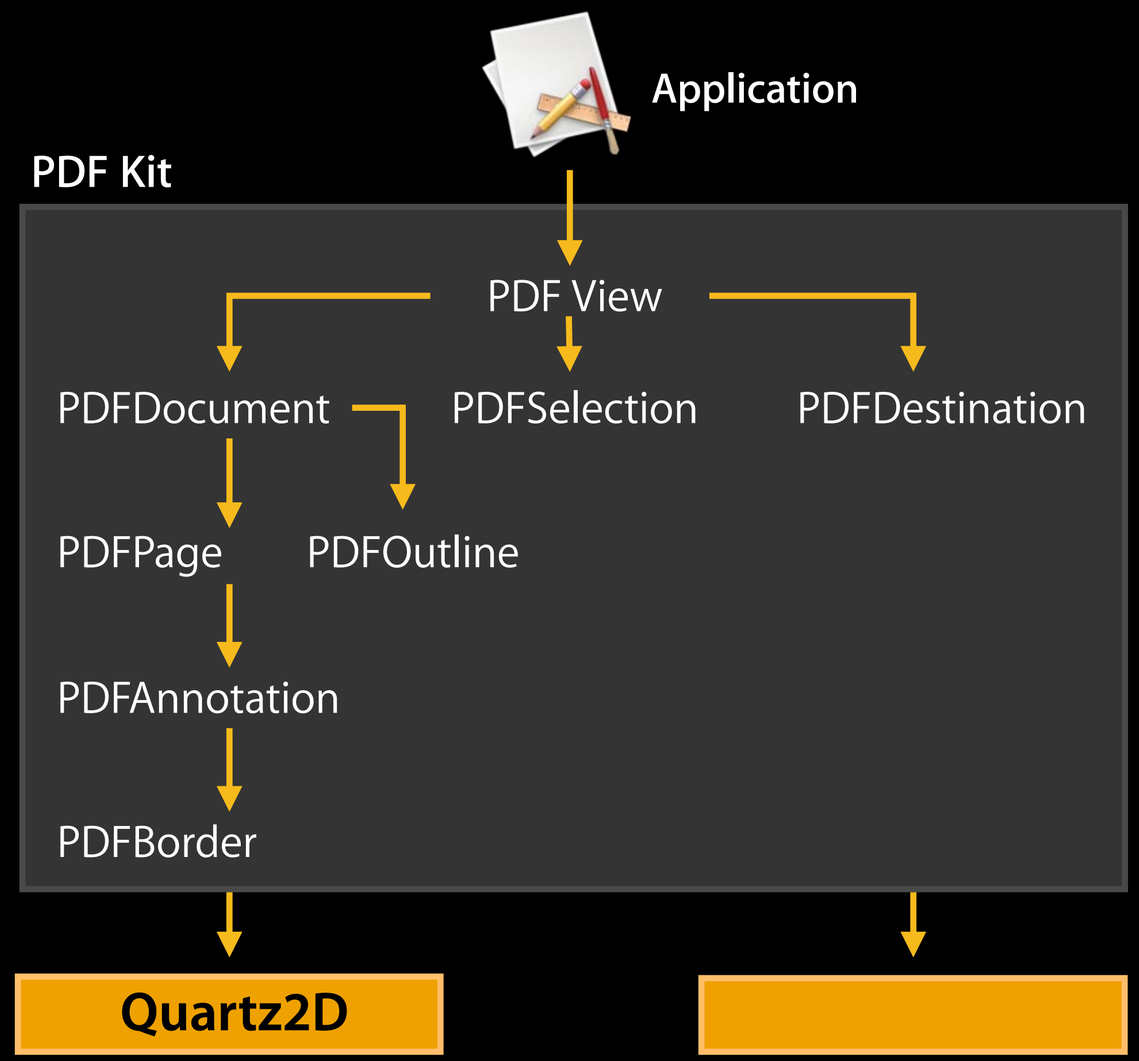
PDF Kit

John Calhoun
Software Engineer

http://developer.apple.com/wwdc2005/

PDF Kit

- Used in Preview and Safari
- Part of Quartz.framework
- Suite of Cocoa classes
- Calls through to Quartz 2D

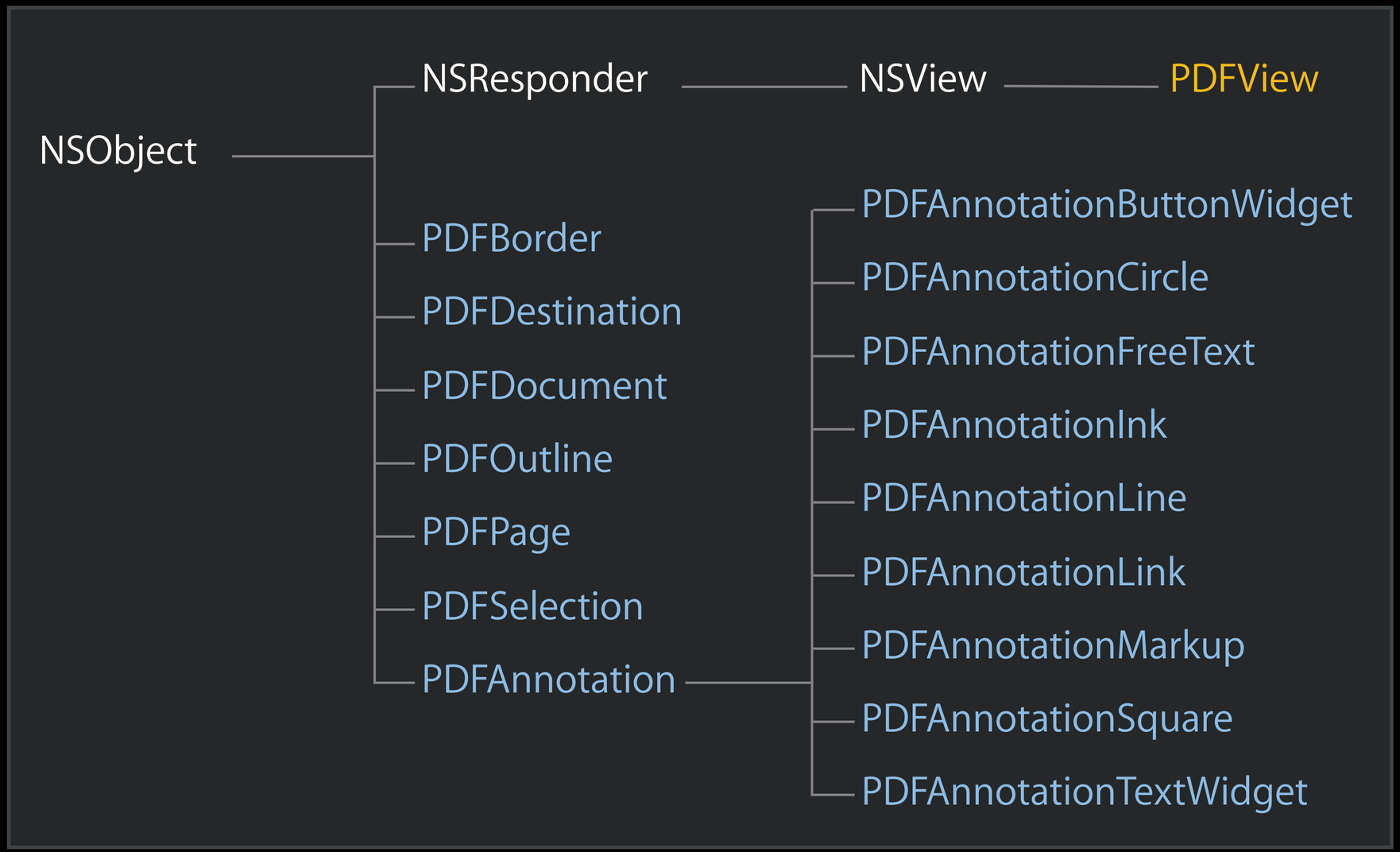


Demo

Safari and PDF Kit

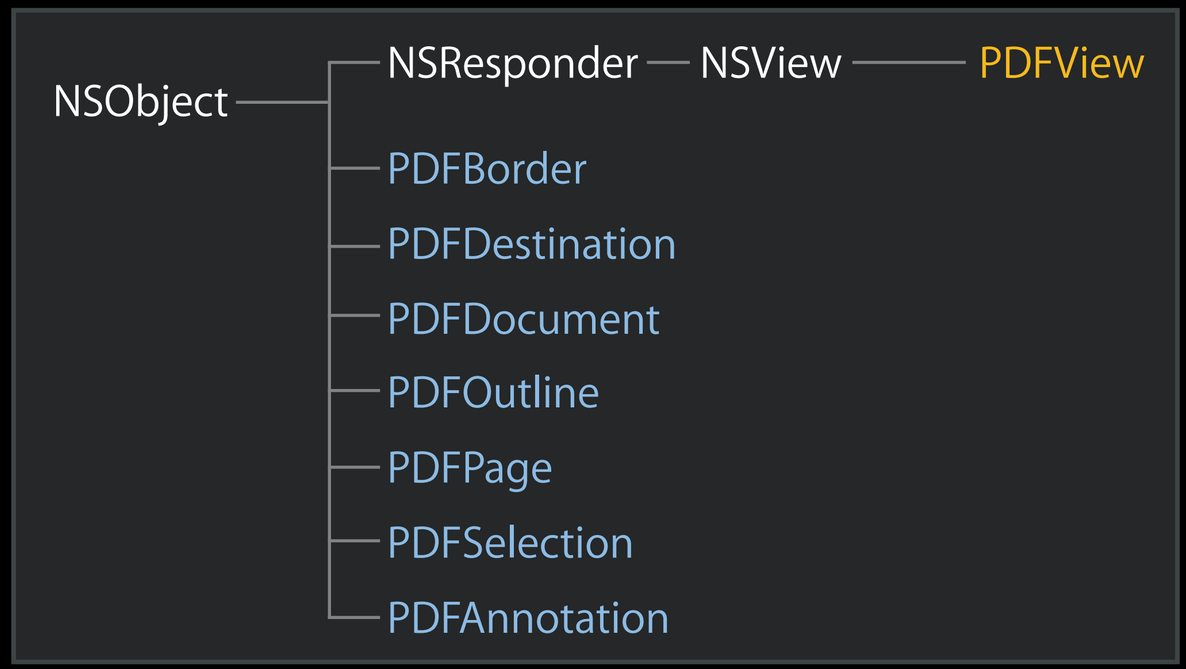
http://developer.apple.com/wwdc2005/

Class Hierarchy



High-Level and Low-Level

- One high-level view class
  - PDFView
- Several utility or low-level classes
  - PDFDocument
  - PDFPage
  - Etc.



Demo

PDF Linker—Opening a PDF

http://developer.apple.com/wwdc2005/

PDFView—Associating a PDFDocument

```
PDFDocument   *myPDFDoc;

// Create PDFDocument.
myPDFDoc = [[PDFDocument alloc] initWithURL:
     [NSURL fileURLWithPath: myFileName]];

// Set document.
[myPDFView setDocument: myPDFDoc];

// We can release since PDFView retains.
[myPDFDoc release];
```

PDFDocument—Attributes

- Number of pages
- Pages
- Document information
- Searching
- Outline
- Document saving

Demo

PDF Linker—Navigation

http://developer.apple.com/wwdc2005/

PDFView—Page Navigation

```
// Simple page navigation.
[myPDFView goToNextPage: sender];
[myPDFView goToPreviousPage: sender];

// Page history navigation.
[myPDFView goBack: sender];
[myPDFView goForward: sender];
enable = [myPDFView canGoBack];
enable = [myPDFView canGoForward];

// Go to a specific page.
PDFPage    *newPage;

newPage = [[myPDFView document] pageAtIndex: index - 1];
[myPDFView goToPage: newPage];
```

PDFView—Notifications

```
// Listen for PDFViewPageChangedNotification.
[[NSNotificationCenter defaultCenter] addObserver: self
     selector: @selector(updatePageNumberField:)
     name: PDFViewPageChangedNotification
     object: myPDFView];
```

- Other PDFView notifications
- PDFDocument notifications

Demo

PDF Linker—Display Options

http://developer.apple.com/wwdc2005/

PDFView—Display Options

```
// Tell PDFView to display single pages.
[myPDFView setDisplayMode: kPDFDisplaySinglePage];
```

- Display boxes
- Page breaks
- Scaling
- Greeking
- Anti-aliasing

Demo

PDF Linker—Outlines

http://developer.apple.com/wwdc2005/

PDFOutline—Root Outline Object

```
PDFOutline    *myOutline;

// Get outline from PDFDocument (note: may return NULL).
myOutline = [[myPDFView document] outlineRoot];
if (myOutline)
   [myOutline retain];
```

PDFOutline—Attributes

- Number of children
- Children
- Label
- Destination

Demo

PDF Linker—Searching

http://developer.apple.com/wwdc2005/

PDFDocument—Search Method 1

```
// Find next instance of searchString (note: may return NULL).
// To find previous instance “or-in” NSBackwardsSearch with myOptions.
selection = [[myPDFView document] findString: searchString
     fromSelection: [myPDFView currentSelection]
     withOptions: myOptions];

// If an instance found, set as current selection and scroll there.
if (selection)
{
   [myPDFView setCurrentSelection: selection];
   [myPDFView scrollSelectionToVisible: self];
}
```

PDFDocument—Search Method 2

```
// Initiate search (will search forward, non-literal, case-insensitive).
[[myPDFView document] beginFindString: searchString
     withOptions: NSCaseInsensitiveSearch];

// PDFDocument delegate may implement:
// -[didMatchString: (PDFSelection *) instance]
// Optionally, you could observe PDFDocumentDidFindMatchNotification.
```

PDFSelection—Attributes

- Page(s)
- Bounds for page
- Nearest PDFOutline item
- Adding
- Extending
- String (and attributed string)

Demo

PDF Linker—PDFView Subclassing

http://developer.apple.com/wwdc2005/

PDFView—drawPage

```
// Override drawPage.
- (void) drawPage: (PDFPage *) pdfPage
{
   // Let PDFView do most of the hard work.
   [super drawPage: pdfPage];

   // When we are in edit mode, append our own drawing....
   if (linkEditMode == YES)
   {
      // Drawing code here....
      ....
   }
}
```

PDFPage—Attributes

- Document
- Bounds
- Rotation
- Annotations
- Text
- Draw method

PDFPage—Annotations

```
// Inside -[drawPage:] - loop over annotations for page looking for links.
NSArray    *allAnnotations;

// Get annotation array from page (note: may return NULL).
allAnnotations = [pdfPage annotations];
if (allAnnotations)
{
   unsigned int            i;

   // Walk through array of annotations.
   for (i = 0; i < [allAnnotations count]; i++)
   {
      // Handle individual annotations...
      ...
   }
}
```

PDFAnnotation—Attributes

- Type
- Page
- Bounds
- Draw method
- Various type-specific attributes

PDFAnnotation—Type

```
// While looping over annotations (index is i).
PDFAnnotation      *annotation;

// Get a single annotation from array.
annotation = [allAnnotations objectAtIndex: i];

// Test annotation type to see if it is a link annotation.
if ([[annotation type] isEqualToString: @”Link”] == YES)
   {
      // Special drawing code for the link annotation...
      ...
   }
}
```

Demo

PDF Linker—Creating Link Annotations

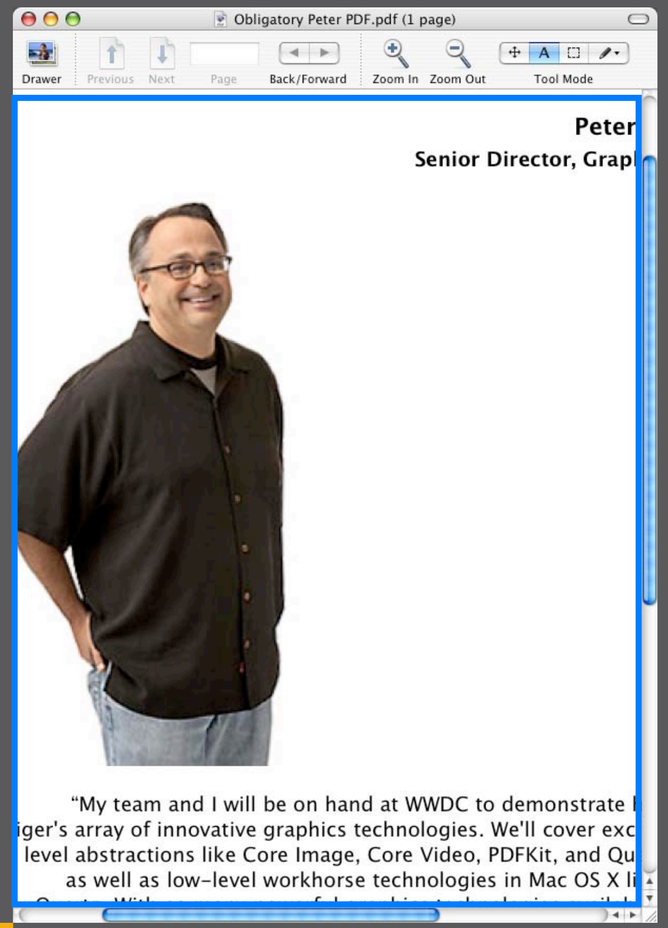
http://developer.apple.com/wwdc2005/

PDFAnnotationLink—Creating

```
// Get pages covered by selection.
allPages = [selection pages];

// Get first page covered by selection.
onePage = (PDFPage *)[allPages objectAtIndex: 0];

// Get bounds covered on first page.
bounds = [selection boundsForPage: onePage];

// Create the link annotation.
newLink = [[PDFAnnotationLink alloc] initWithBounds: bounds];

// Add to page.
[onePage addAnnotation: newLink];
```

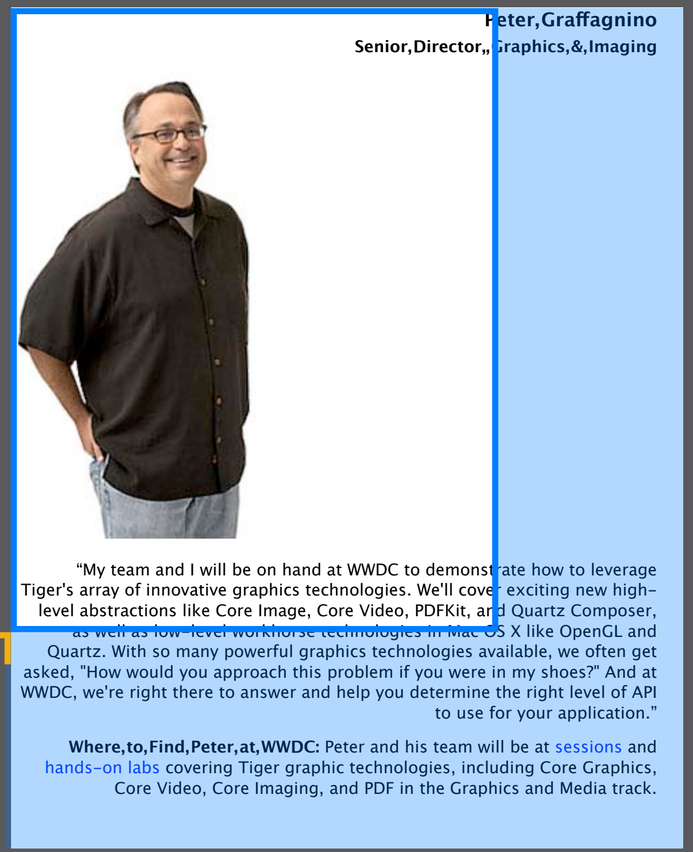
“Page Space” vs. “View Space”

View Space



origin

Page Space



origin

PDFDestination—Creating

```
// What page is at the top-left corner of the PDFView?
topLeftPt = NSMakePoint(0, [myPDFView frame].size.height]);
topLeftPage = [myPDFView pageForPoint: topLeftPt nearest: YES];

// Get top left point in “page space” (note: we re-use topLeftPt).
// Note 2: really should pin point to bounds of topLeftPage.
topLeftPt = [myPDFView convertPt: topLeftPt toPage: topLeftPage];

// Now create a PDFDestination for the given page and point.
destination = [[PDFDestination alloc] initWithPage: topLeftPage
     atPoint: topLeftPt];

// Set destination.
[activeLink setDestination: destination];
```

PDF Workflows

Richard Blanchard
Software Engineer

http://developer.apple.com/wwdc2005/

Great PDF Apps

- Full-blown applications using
  - PDF Kit
  - Quartz
  - Cocoa

Don’t Swing for the Fences

- Often times you don’t need a full-blown application
  - Command line tools
  - Automator plug-ins

High-Level PDF

- Python scripts to create PDF
- Automator actions to analyze PDF
  - Quartz PDF Scanner
- PDF Workflow from the print dialog
- PDF Workflow APIs for applications

Writing PDF Scripts

Python

- Panther and Tiger include Python bindings for Quartz
  - Simple wrapper for Quartz calls
  - Wrappers for some higher level function
    - HTML, RTF, variety of image formats

Quartz Python

- Great for experimenting with Quartz
- Great for writing tools
- Great for the core of an Automator action
  - Wrap as a Terminal action

Python and Quartz

- For a description of the Python calls see
  - /Developer/Examples/Quartz/Python/API-SUMMARY

Example PostScript RIP

Part I: PS to PDF

```
#!/usr/bin/python
# Usage: psrip ps-file tiff-base
#
import sys
from CoreGraphics import *

psProvider = CGDataProviderCreateWithFilename(sys.argv[1])
pdfConsumer = CGDataConsumerCreateWithFilename(sys.argv[2])
converter = CGPSConverterCreateWithoutCallbacks()
success = converter.convert(psProvider, pdfConsumer)
```

Example PostScript RIP

Part II: PDF to TIFF

```
if success :
  pdf = CGPDFDocumentCreateWithProvider(
          CGDataProviderCreateWithFilename(sys.argv[2])

  for i in range(1, pdf.getNumberOfPages() + 1) :
    mediaBox = pdf.getMediaBox(i)
    ctx = CGBitmapContextCreateWithColor(
            mediaBox.size.width, mediaBox.size.height,
            CGColorSpaceCreateDeviceRGB(), (0, 0, 0, 1))

    ctx.drawPDFDocument(mediaBox, pdf, i)
    path = "%s.%d.tiff" % (sys.argv[2], i)

    ctx.writeToFile(path, kCGImageFormatTIFF, "")
```

Python Libraries

- Large, existing base of Python code and libraries
  - Vaults of Parnassus
    - http://www.vex.net/parnassus/

Quartz Integration

Use Quartz bindings to integrate Python libraries and products with Mac OS X

- Server side content creation
  - Zope/Plone
- Python graphics libraries
  - PyChart

PyChart

- “A Python library for creating high quality Encapsulated Postscript, PDF, PNG, or SVG charts. It currently supports line plots, bar plots, range-fill plots, and pie charts”
- GPL by Yasushi Saito
  - http://home.gna.org/pychart/

PyChart

- Renders vector charts
- Existing library can create
  - PDF and PostScript
    - Can import these with Quartz bindings
  - Much rather have PyChart render directly into a Quartz context

PyChart Extended

- Making PyChart render into a Core Graphics context
  - Add a ‘canvas’ to PyChart
    - quartzcanvas.py
  - Trivial due to the shared PDF/PS imaging model

Quartz Canvas

```
def __init__(self, fname, cgContext = None):
    basecanvas.T.__init__(self)

    if cgContext == None:
       self.__context = CGPDFContextCreateWithFilename
                    (fname, CGRectMake (0, 0, 612, 792))
        self.__context.beginPage
                    (CGRectMake (0, 0, 612, 792))
        self.__ourContext = True
    else:
        self.__context = cgContext
        self.__ourContext = False

    self.__colorSpace = None
    self.__graySpace = None
    self.__line_style = None
```

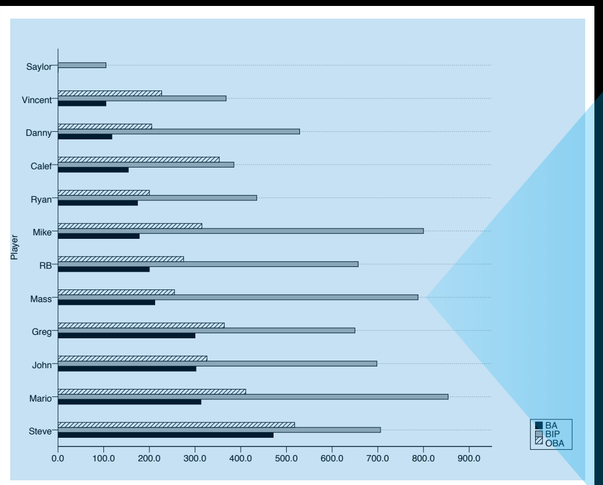
Quartz Canvas

```
def moveto(self, x, y):
  self.__context.moveToPoint(x, y)

def lineto(self, x, y):
  self.__context.addLineToPoint(x, y)

def set_fill_color(self, color):
  self.__context.setFillColorWithColor(self._getCGColor(color))

def fill(self):
  self.__context.fillPath()

def stroke(self):
  self.__context.strokePath()

def gsave(self):
  self.__context.saveGState()
```

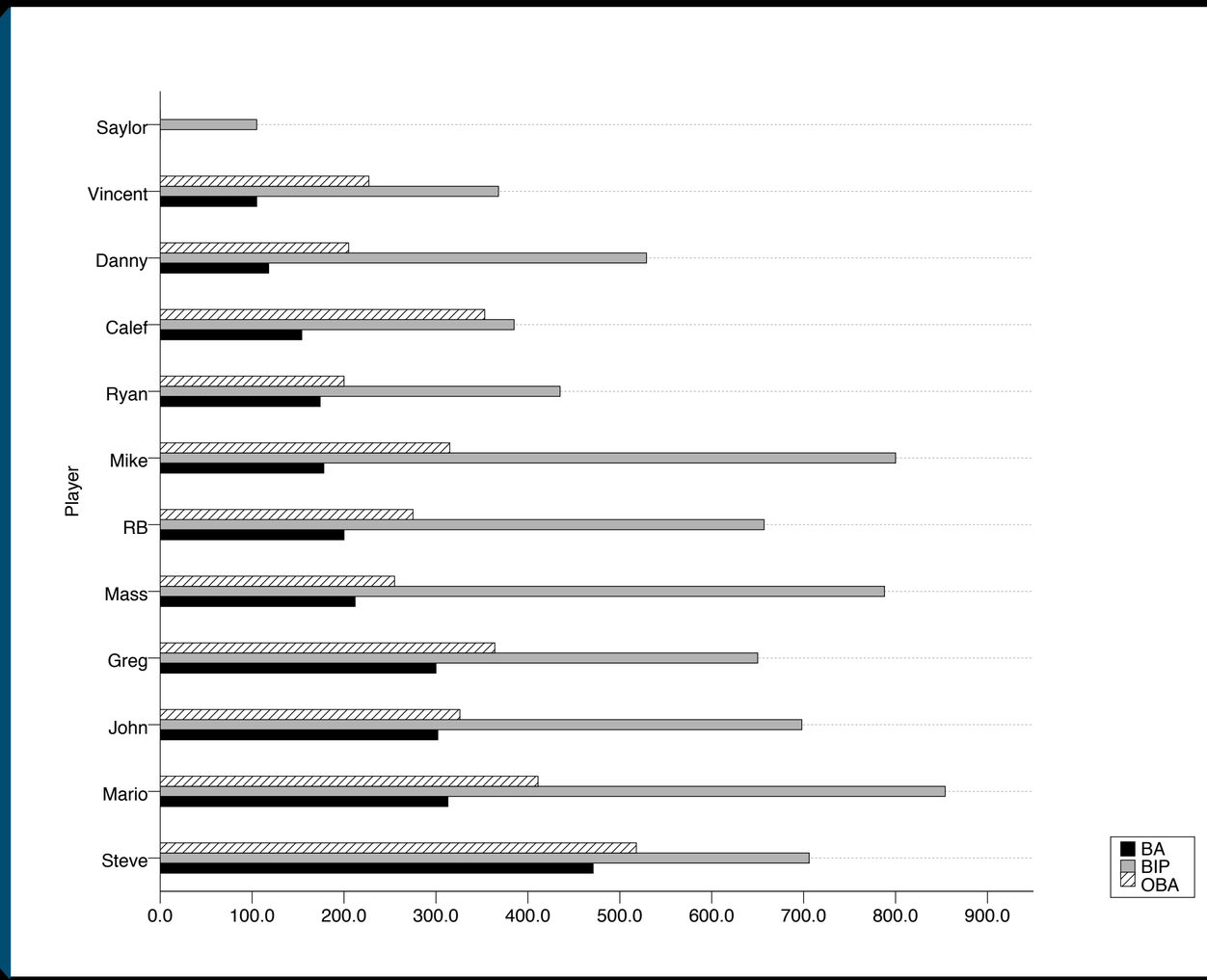
Mixing Quartz and PyChart

- Once PyChart can render to a Quartz context
  - Charts can be mixed with other Quartz binding drawing
    - For example a chart and a table

Chart and Table

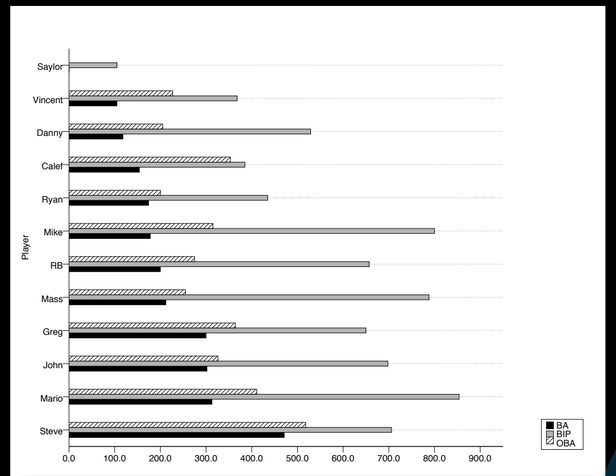


Player	BA	BIP	OBA
Steve	471.0	706.0	518.0
Mario	313.0	854.0	411.0
John	302.0	698.0	326.0
Greg	300.0	650.0	364.0
Mass	212.0	788.0	255.0
RB	200.0	657.0	275.0
Mike	178.0	800.0	315.0
Ryan	174.0	435.0	200.0
Calef	154.0	385.0	353.0
Danny	118.0	529.0	205.0
Vincent	105.0	368.0	227.0
Saylor	0.0	105.0	0.0



PyChart

Chart and Table



Player	BA	BIP	OBA
Steve	471.0	706.0	518.0
Mario	313.0	854.0	411.0
John	302.0	698.0	326.0
Greg	300.0	650.0	364.0
Mass	212.0	788.0	255.0
RB	200.0	657.0	275.0
Mike	178.0	800.0	315.0
Ryan	174.0	435.0	200.0
Calef	154.0	385.0	353.0
Danny	118.0	529.0	205.0
Vincent	105.0	368.0	227.0
Saylor	0.0	105.0	0.0

Player	BA	BIP	OBA
Steve	471.0	706.0	518.0
Mario	313.0	854.0	411.0
John	302.0	698.0	326.0
Greg	300.0	650.0	364.0
Mass	212.0	788.0	255.0
RB	200.0	657.0	275.0
Mike	178.0	800.0	315.0
Ryan	174.0	435.0	200.0
Calef	154.0	385.0	353.0
Danny	118.0	529.0	205.0
Vincent	105.0	368.0	227.0
Saylor	0.0	105.0	0.0

HTML Rendered Table

Main

```
data = chart_data.read_csv(file)

mediaBox = CGRectMake(0, 0, 8.5 * 72, 11 * 72)
ctx = CGPDFContextCreateWithFilename("out.pdf", mediaBox);
ctx.beginPage(mediaBox);

drawBarGraph(ctx, data)
drawTable(ctx, data)

ctx.endPage()
ctx.finish()
```

Drawing the Chart

```
def drawBarGraph(context, data) :
    columns = data[0]
    data = data[1:]

    ar = area.T(loc = (1 * 72, 5 * 72), size=(6*72, 5.5*72),
            y_coord = category_coord.T(data, 0),
             x_axis = axis.X(label=""),
             y_axis = axis.Y(label=columns[0]))

    chart_object.set_defaults(bar_plot.T, direction="horizontal",
data=data)
    for i in range(1,len(columns)) :
        ar.add_plot(bar_plot.T(hcol = i, cluster=(i-1, len
(columns)-1), label=columns[i]))

    can = canvas.init(format="quartz", cgContext=context)
    ar.draw(can)
```

Drawing the Table

```
def drawTable(context, data) :
    html = '<html><TABLE border="1"><TR>'
    tableHeader = data[0]
    data = data[1:]

    for title in tableHeader:
        html = html + '<TH>' + title + '</TH>'
    html = html + '</TR>'

    for row in data :
        html = html + '<TR>'
        for col in row :
            html = html + '<TD>' + str(col) + '</TD>'
        html = html + '</TR>'
    html = html + '</TABLE></html>'

    rect = CGRectMake(1 * 72, 0.5 * 72, 6 * 72, 4 * 72)
    context.drawHTMLTextInRect(CGDataProviderCreateWithString
(html),rect, 10)
```

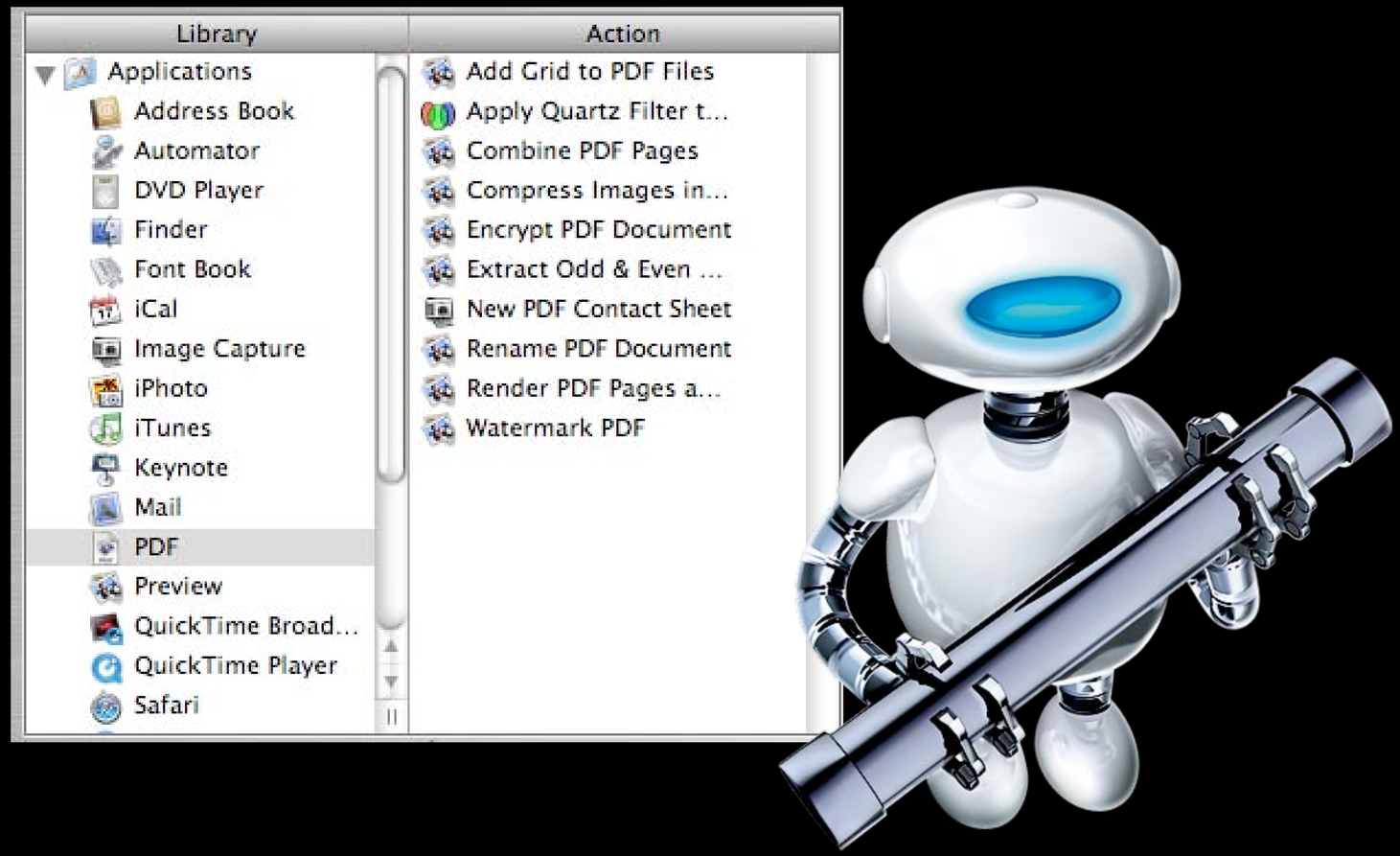
Quartz Python

PyChart is just an example

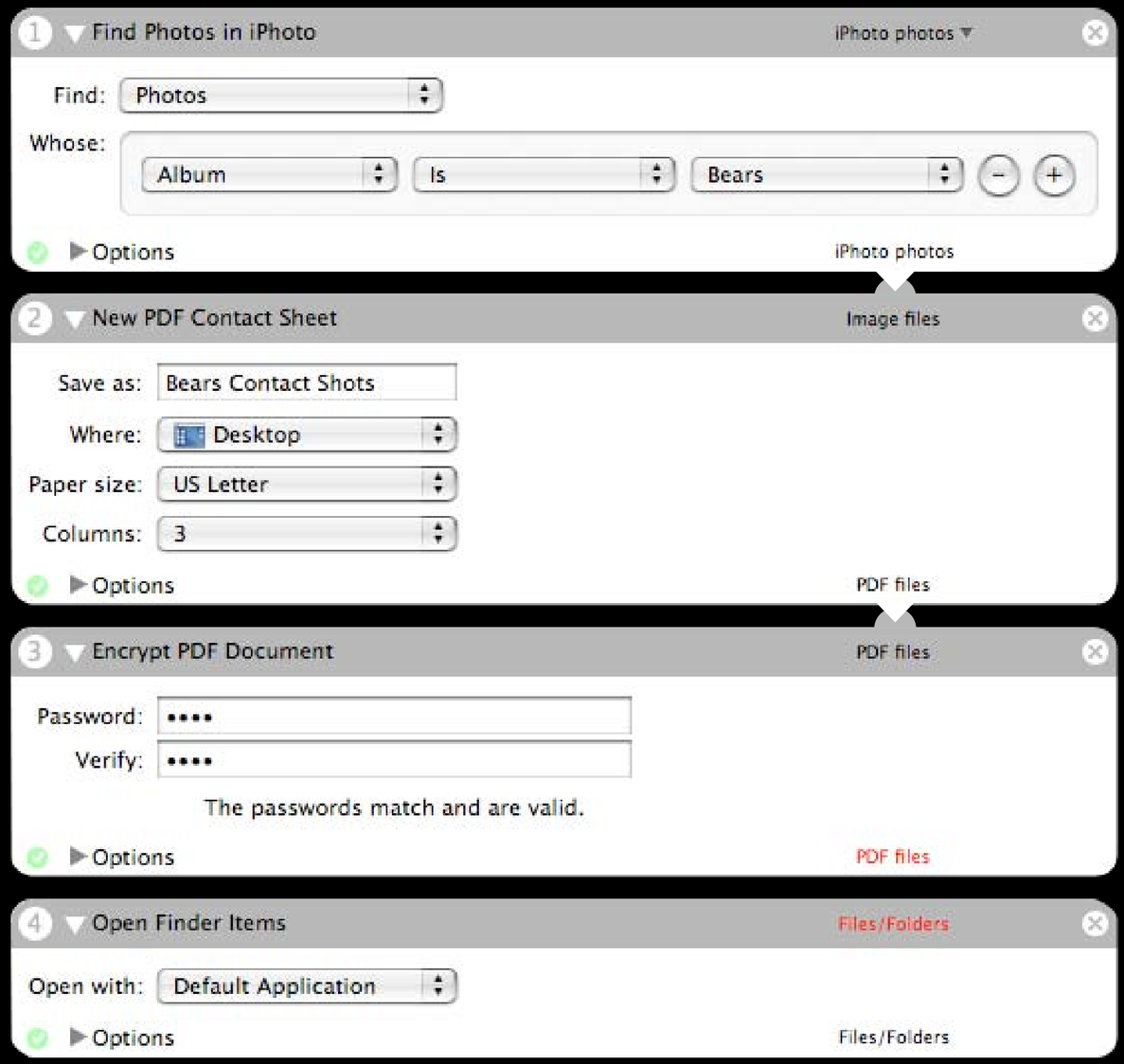
- Other Python libraries can be integrated in the same way
- Common PostScript/PDF imaging model makes this easy
- High quality vector PDF output is the result

PDF Automator Actions

Automator ships with ten PDF Actions



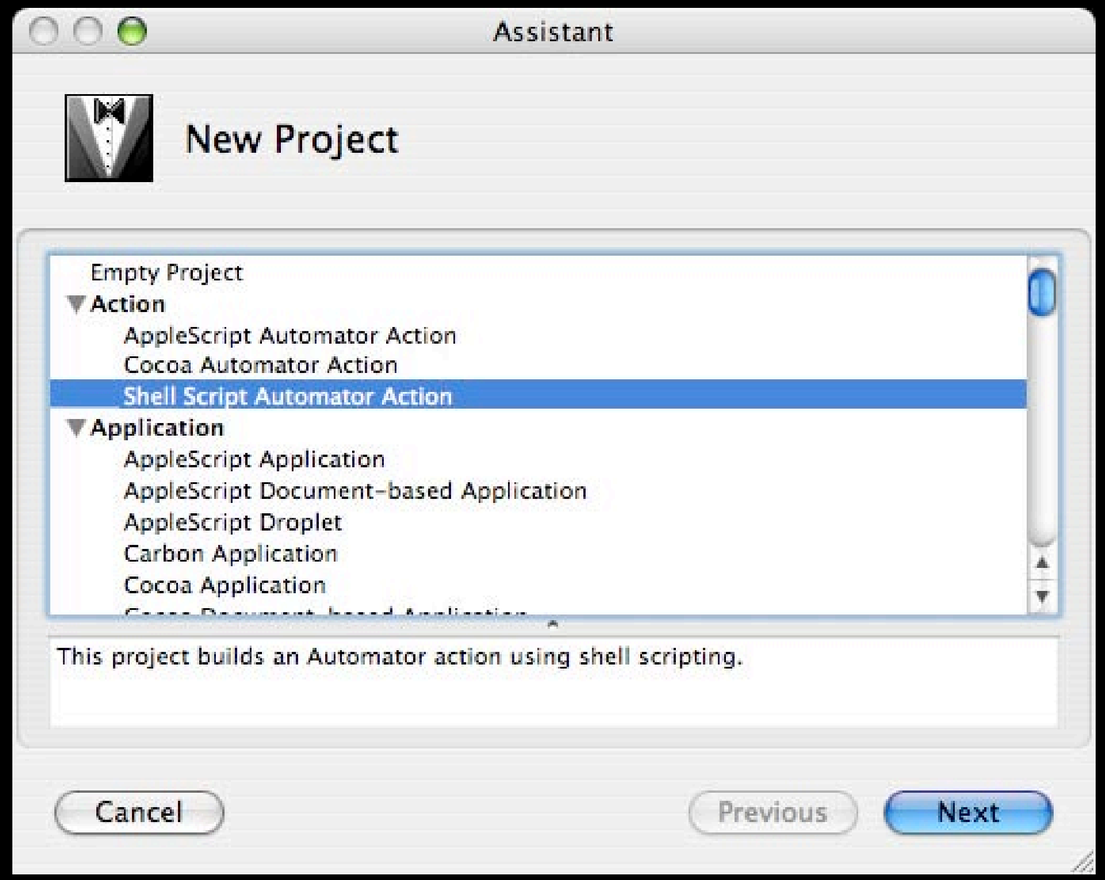
Automator Workflow



Easy to Make More PDF Actions

- Use Python script in a Shell Action
- Write a Cocoa action using Quartz APIs

A Shell Action



Xcode 2.1

Chart and Table Action

Wrap chart and table python script in a shell action

- Easy
  - Put the script in the skeleton
  - Edit the description of the action
  - Tweak the nib

Writing Cocoa Actions

- Subclass AMAction and implement

```
- (id)runWithInput:(id)input
        fromAction:(AMAction *)anAction
             error:(NSDictionary **)errorInfo
```

- Use Quartz to access and manipulate the PDF file

PDF Scanning API

- New in Tiger
- Quartz client can provide function to be invoked based on PDF operators
- Particularly useful for analyzing the content of a PDF

Using the Scanning API

- Start with a PDF document
  - `pdf = CGPDFDocumentCreateWithURL(url);`
- Get a page from the document
  - `page = CGPDFDocumentGetPage(pdf, i);`
- Get the page's content stream
  - `stream = CGPDFContentStreamCreateWithPage(page);`

Scanning the Stream

- Define the operator table
  - `opTable = CGPDFOperatorTableCreate()`
- Declare your callbacks
  - `CGPDFOperatorTableSetCallback(opTable, "Do", myDoCallback);`
- Create a scanner with that operator table
  - `scanner = CGPDFScannerCreate(stream, opTable, self);`
  - `success = CGPDFScannerScan(scanner);`

Operator Callbacks

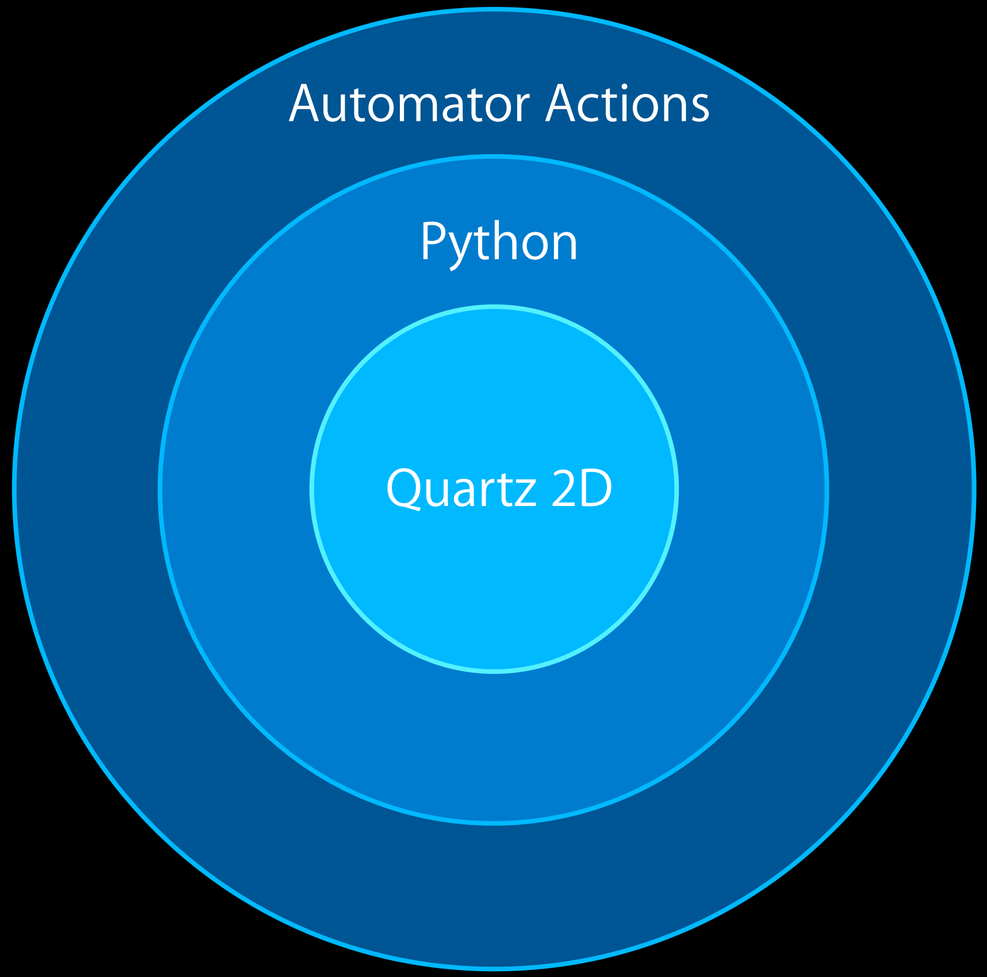
- Once inside your callback
  - Retrieve operator parameters
  - Get XObjects
  - Record information about the PDF

PDF Image Counter

- Create a Cocoa Automator action
- Use PDF stream and scan APIs
  - Count images per page
- Generate CSV summary of image totals

Putting it All Together

- In Automator create a workflow
  - PDF to image summary CSV
  - CSV to PDF chart
  - Open PDF
- Save as Printing Plugin
- Invoke workflow from the print dialog



Demo

http://developer.apple.com/wwdc2005/

PDF Workflow API

- Applications can execute the PDF Workflow items directly
- To get the list of available workflow items
  - `OSStatus PMWorkflowCopyItems(CFArrayRef * workflowItems)`
  - Each workflow item is represented by a CFURLRef

PDF Workflow API

- A PDF workflow item starts with a PDF and does something with it or to it
- To execute a specific PDF workflow URL

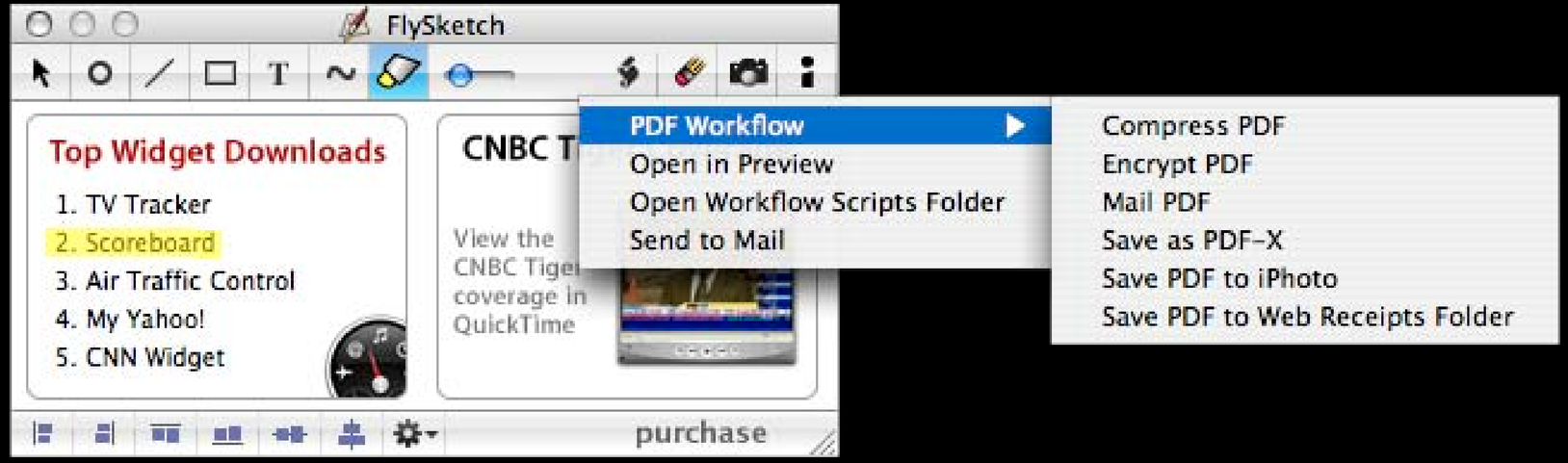
```
OSStatus PMWorkflowSubmitPDFWithOptions
         (CFURLRef workflowItem,
          CFStringRef   title,
          const char *  options,
          CFURLRef      pdfFile)
```

PDF Workflow API

- Workflow items that are supported in Tiger
  - Folder
  - Folder Alias
  - UNIX tool
  - AppleScript
  - Automator workflow
  - Application
  - Quartz filter

FlySketch Example

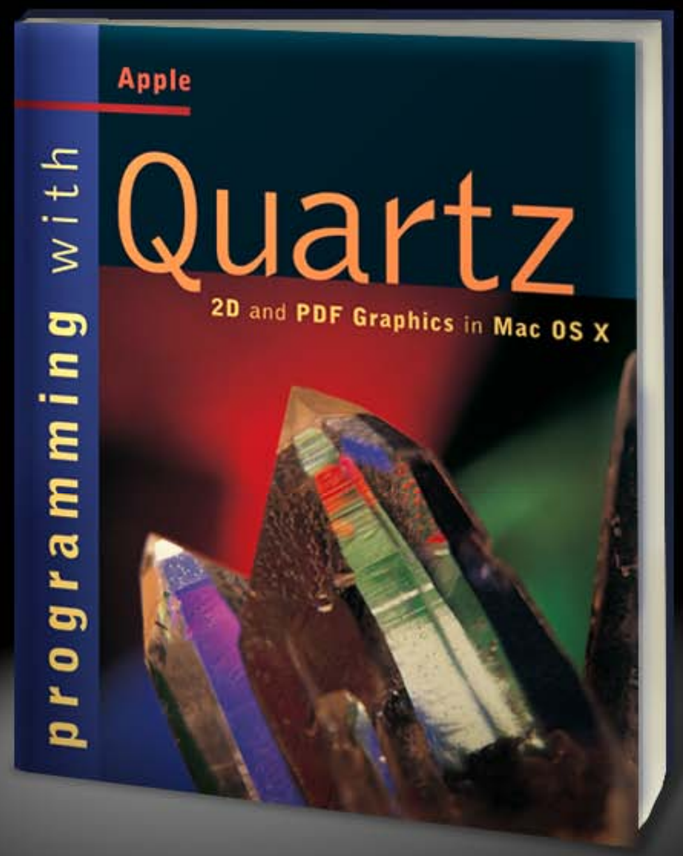
- FlySketch 1.1a1 from Flying Meat Software
- Using the PDF Workflow API applications can offer user extensible PDF processing



High-Level PDF

- Use Python bindings to
  - Experiment with Quartz capabilities
  - Write simple PDF scripts
- Use Automator
  - Wrap scripts into PDF actions
  - Create Cocoa actions
- Use the PDF Workflow API

Coming Soon From Morgan Kaufmann…



More info and sample chapter
http://books.elsevier.com/mk/?isbn=0123694736

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Moving from QuickDraw to Quartz 2D	Session 215 Friday 9:00AM
Optimal 2D Application Graphics	Session 216 Friday 10:30AM
Graphics and Media Feedback Forum	Session 217 Friday 2:00PM

Who to Contact

Travis Brown
Graphic and Imaging Evangelist
travis@apple.com

Q&A

Travis Brown
Graphic and Imaging Evangelist
travis@apple.com

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference